3DNow!™ Instruction Set and AMD Athlon™ Processor Overview

David Kaplowitz

Senior Member Technical Staff

AMD Advanced Architecture Labs

AMD

SIGGRAPH Los Angeles

Agenda

3DNow!™ Instruction Set

- AMD-K6®-2 and AMD-K6®-III
- AMD Athlon™ Processor extensions

AMD Athlon™ Processor programming

- Architecture
- Low-level optimization

AMD

SIGGRAPH Los Angeles

3DNOW!™ Instruction Set

History

- Introduced in AMD-K6®-2 in 1H98
  - *SIMD floating point to accelerate geometry pipeline*
- Supported in AMD-K6®-III in 2H98
  - *Addition of internal L2 cache*
- Instruction set extended in Athlon™ (introduced yesterday!)
  - *Mostly integer MMX™ with a few SIMD FP instructions*

AMD

MMX™ is a trademark of Intel Corporation

MMX™ is a trademark of Intel Corporation

SIGGRAPH Los Angeles

3DNOW!™ Instruction Set

Analogous to MMX™ instructions

- Operate on 64-bit MMX registers
- New packed data type: two 32-bit floating point values
- Can be freely intermixed with MMX instructions
  - *MOVD, MOVQ and PUNPCKLDQ among others, are often used*
- Check bit 31 of Extended Feature Flags

3DNOW!™ Instruction Set

Standard two-operand instruction format:

- mmreg1 ← mmreg1 OP mmreg2/mem64
- PFADD mm0, mm1
  - *mm0[63:32] ← mm0[63:32] + mm1[63:32]*
    mm0[31:0] ← mm0[31:0] + mm1[31:0]
- PFADD mm0, [EAX]
  - *mm0[63:32] ← mm0[63:32] + [EAX][63:32]*
    mm0[31:0] ← mm0[31:0] + [EAX][31:0]

AMD

SIGGRAPH Los Angeles

3DNOW!™ Instruction Set

Addition and Subtraction

- PFADD
  - *Op1 ← Op1 + Op2*
- PFSUB
  - *Op1 ← Op1 - Op2*
- PFSUBR
  - *Op1 ← Op2 - Op1*

AMD

SIGGRAPH Los Angeles

3DNOW!™ Instruction Set

Accumulation and Multiplication

- PFACC
  - *Adds high and low halves of each operand, packs two results together*
  - *Useful for dot products*
- PFMUL
  - *Op1 ← Op1 * Op2*

3DNOW!™ Instruction Set

Comparison and Conversion

- PFCMPGT, PFCMPEQ, PFCMPGE
  - *Produces two all 0's or all 1's masks*
- PFMIN, PFMAX
  - *Returns the lower/higher of two operands in each half*
- PF2ID, PI2FD
  - *Conversion to/from 32-bit signed integers*

3DNOW!™ Instruction Set

Reciprocal and reciprocal square root approx

- PFRCP, PFRSQRT
- Table lookups to produce 14-bit reciprocal and 15-bit reciprocal square root approximations, fully pipelined
- Scalar Operation - lookup of low operand only, approximation is duplicated in high and low halves
- Particularly useful in lighting calculations - gives sufficient accuracy in the majority of cases

3DNOW!™ Instruction Set

Full precision reciprocal and reciprocal sqrt

- PFRCPIT1, PFRCPIT2, PFRSQIT1
- Yields full 24-bit reciprocal/reciprocal square root via one iteration of Newton-Raphson algorithm, fully pipelined
- Reciprocal: PFRCP, PFRCPIT1, PFRCPIT2
- Reciprocal square root: PFRSQRT, PFMUL, PFRSQIT1, PFRCPIT2

3DNOW!™ Instruction Set

Prefetch

- PREFETCH and PREFETCHW
- Enables data cache fills to be scheduled ahead of data use
  - *Maximizes overlap of computation with cache fills*
- Use on predictable data streams
  - *Example: processing an array of triangle vertices*
- PREFETCHW indicates intent to modify data

3DNOW!™ Instruction Set

x87 ↔ MMX™/3DNow!™ Context Switch

- FEMMS
- Enables faster switch between executing MMX/3DNOW! and x87 FPU instructions on K6-2 and K6-III
- Like EMMS, plus sets all register values to undefined
- Use in place of EMMS at end of MMX/3DNOW! routines
- Also use at beginning of MMX/3DNOW! routines
  - *Gives faster x87 to MMX/3DNOW! switch*

AMD

SIGGRAPH Los Angeles

3DNOW!™ Instruction Set

Integer Instructions

- PAVGUSB
  - *MMX instruction for motion compensation for video decode*
  - *Computes eight byte-sized (a + b + 1) / 2*
- PMULHR
  - *Like MMX PMULH, but rounds versus truncates low half of product*

AMD

SIGGRAPH Los Angeles

Instruction set extensions for Athlon™

Enhanced 3DNow!™ and MMX™

- Total of 24 new instructions
- SIMD FP instructions
  - *Check bit 30 of Extended Feature Flags*
  - *Useful for traditional DSP such as audio and modem*
- SSE MMX™ instructions
  - *Check bit 22 of Extended Feature Flags*
  - *Some are useful for graphics or video*

AMD

SIGGRAPH Los Angeles

Enhanced 3DNOW!™

Accumulation Instructions

- SIMD FP - useful for audio and modem
- PFNACC
  - *Negative accumulate: low - high for each operand*
- PFPNACC
  - *Positive-negative accumulate: negative accumulate for Op1, positive accumulate for Op2*

Enhanced 3DNOW!™

Data conversion

- PI2FW and PF2IW
  - *Data conversion WORD to/from FLOAT.*
  - *Lighting calculations*
- PSWAPD - Packed Double Word Swap
  - *Endian conversion*

Enhanced MMX™

Prefetch

- PREFETCHNTA, PREFETCHT0, PREFETCHT1, PREFETCHT2
- Affects which cache levels data is prefetched into
- All currently implemented the same as PREFETCH

Enhanced MMX™

Write control

- MOVNTQ and MASKMOVQ
  - *Instructions defined as cache write-around but implemented as write-back*
- SFENCE
  - *Flush all write buffers (for serialization)*

Enhanced MMX™

Video Instructions

- PSADBW
  - *Sum of absolute byte differences*
  - *Useful for motion estimation during video encode*
- PAVGB
  - *Same as PAVGUSB*
  - *Useful for motion compensation during video decode*

Enhanced MMX™

Others

- PINSRW - Insert word from integer register
- PEXTRW - Extract word into integer register
- PMOVMSKB - Move mask to integer register
- PSHUFW - Word shuffle
- PMINUB - Unsigned byte minimum
- PMAXUB - Unsigned byte maximum
- PAVGW - Unsigned word average

AMD

SIGGRAPH Los Angeles

Enhanced MMX™

Others

- PMINSW - Signed word minimum
- PMAXSW - Signed word maximum
- PMULHUW - Unsigned word multiply unsigned high
- FXSAVE/FXRSTOR - extended context save and restore (only for OS writers)

Enhanced Integer Instructions

Conditional MOV

- CMOV and FCMOV

AMD Athlon™ Design Goals

Designed for high performance at high clock rate

- Deep scheduler and aggressive reordering smoothes out most code
- Easy guidelines to maximize performance with few end cases
- Scalable beyond 1 GHz

AMD

SIGGRAPH Los Angeles

AMD Athlon™ Architecture

Enhanced microarchitecture & system features

- Today
  - *200 MHz frontside bus*
  - *flexible backside L2 cache*
  - *128-KB L1 with 64-byte cache line*
  - *aggressive write combining,*
  - *deep internal buffering*

AMD

SIGGRAPH Los Angeles

AMD Athlon™ Architecture

Enhanced microarchitecture & system features

- Future
  - *support for AGP 4X*
  - *DDR SDRAM*
  - *RAMBUS*
  - *SMP*

AMD

SIGGRAPH Los Angeles

AMD Athlon™ Processor Architecture

- Decode up to three x86 instructions per cycle
- Dynamic scheduling with speculative, out-of-order execution
- 3 Superscalar, out-of-order integer pipelines
  - *each with an address generation unit and execution unit*
- 3 Superscalar, out-of-order multimedia pipelines
  - *single cycle throughput*
  - *killer x87 floating-point performance*
  - *Zero overhead x87 and 3DNow!/MMX context switch*

AMD

SIGGRAPH Los Angeles

AMD Athlon™ Processor Architecture

- 64-KB L1 I-cache & 64-KB L1 D-cache
  - *each 2-way Set Associative*
- Dual-ported D-cache
  - *multi-banking allows concurrent access by 2 load/stores*
- Flexible high-speed 64-bit backside L2 cache controller
  - *supports 512-KB to 8-MB with programmable data rates*

AMD Athlon™ Processor Architecture

- High-speed 64-bit system interface
  - *first mainstream systems to have a 200MHz Bus*
  - *significant headroom for the future*
- Deep internal buffering to support pipelines and external interfaces
  - *Up to 72 x86 instructions in-flight!!*

AMD

SIGGRAPH Los Angeles

Optimizing for the AMD Athlon™ Processor

Coding Guidelines

- Optimization Guide describes these in the Top Optimizations chapter
- I'll describe some in the next couple of slides
- Not necessary to tweak code down to the cycle
- Also pay attention to dependency chains, resource limitations, and memory bandwidth

Optimizing for the AMD Athlon™ Processor

Some Top Optimizations

- Port floating point intensive sections of code to 3DNow!
- Store to load forwarding
- Branch prediction
- PREFETCH
- DirectPath instructions

Store to Load Forwarding

- Large (32-entry) load/store buffer
- Any data that is stored and loaded again soon:
  - *must be aligned to its natural type*
  - *must be to the same address*
  - *must be the same size*

Store to Load Forwarding

- Avoid:

```
movq [eax], mm0                 ; 64 bit store
.
mov ebx, [eax]                  ; 32 bit load - data not same size
```

- Preferred:

```
movd [eax], mm0                 ; store the low 32-bits
punpckhdq mm0, mm0              ; get upper half in both halves
movd [eax+4], mm0               ; store the upper 32-bits
.
mov ebx, [eax]                  ; store and load are same size
```

Branch Prediction

- Structure code to favor branches not taken
- Avoid dense, successive conditional and unconditional branches
- Avoid data-dependent conditional branches (loop invariant branches are OK)
  - *If your data is totally random, then no branch prediction scheme can predict it correctly!*
  - *Use branch free alternatives whenever possible*

Use PREFETCH / PREFETCHW

Increases performance substantially on Athlon™

- Observations of 20% or more in real world applications
- Multiple outstanding prefetches supported
- Prefetch distance = 200 * DS / C
  - *DS - data stride per loop iteration*
  - *C - cycles for 1 loop iteration when hitting in the L1*
- Use PREFETCHW when you need to modify the data after prefetching it (saves 15-25 cycles)

Use DirectPath Instructions

DirectPath are hardwired instructions, VectorPath must go to microcode

- Athlon™ Decodes up to 3 DirectPath Instructions per Cycle, VectorPath instructions take one or more cycles to decode
- NDA Software Optimization Guide contains list sorted by DirectPath versus VectorPath
- Know what your compiler is doing
  - *AMD working with compiler companies to get support*

AMD Athlon™ Processor Instruction Latencies

Operation	Latency	Throughput
DirectPath Intgr	1	1
MMX ALU	2	1
MMX Mult	3	1
3Dnow! ALU	4	1
3Dnow! Mult	4	1
3Dnow! Recip	3	1

AMD

SI99RAPH
Los Angeles

Summary

- 3DNow! incorporates both SIMD floating point and SIMD integer extensions
- Athlon™ gives fantastic x87 performance and really shows off 3DNow!
- Take an SDK with you